PHYS 643 Problem Set 5

Eve J. Lee

Due: November 12th 2020

In this assignment, you will develop a simple 1-D numerical solver of hydro equations using Python. You are to upload your code to github (`https://github.com/`) for both the ease of sharing and for version control. Make sure your github repository is either public or properly shared with me, and simply post the url on myCourses. Refer to the rubric on myCourses for grading criteria.

Some of the questions require written explanations (to be included in README) for full credit. You are welcome to work with each other but you should write and submit your own code.

There are 4 questions in total:

1. Advection
2. Advection-diffusion
3. 1-D hydro; perturbation
4. Adiabatic shock

Make sure that you are using the correct boundary conditions as required in the questions. If you're using just parts of code you've written before, you still need to cite yourself by stating the course code, year, and the instructor of the class in which you have submitted your work. Remember, self-plagiarism is plagiarism.

If you're replacing your PS5 with a product from Hackathon, the same general grading principle applies. Make sure to indicate that what your code aims to do, specific problem it is solving and how, as well as the names of your team members.

1 To include in your github repository

You should obviously include your Python codes but on top of that, you are to submit a README file that includes the following:

1. Your name (first and last)
2. Version of Python used (2.x or 3.x suffices)
3. List of python code files along with brief descriptions of what they do.
4. For the questions where you're asked to describe what is going on, place your written explanations here. Clearly indicate which code you're referring to.
5. If you have worked closely with your classmates in completing the exercise, list their first and last names. Example: Collaborators: <insert name1>, <insert name2>, etc.

2 Coding style

Start your .py file with a brief description of the code, your name, and the date. If this is a re-submission of a code you wrote in a different class, indicate the course code, year, and the instructor here. Example:

```
"""
Finite differencing example
Advection-Diffusion equation

@author: Eve J. Lee
Jan. 26th 2020
"""
```

Give meaningful and unique names to your variables, and comment each block of code (e.g. “# Setting up the grid”, “# Defining initial conditions”, “# Setting up the plot”, “# Updating the value.”, etc.).

Animate your solution using canvas.draw() of the figure object from matplotlib. Here is one way to set it up:

```
pl.ion()
fig = pl.figure()
x1, = pl.plot(x,f,'ro')
pl.xlim([0,2*Ngrid])
pl.ylim([-0.1,1.0])
fig.canvas.draw()
```

And make sure to include the following for each timestep to update your solution:

```
    x1.set_ydata(f)
    fig.canvas.draw()
    pl.pause(0.001)
```

This is a suggestion, and you’re free to choose your favourite plotting/animation routine as long as they’re clear and visually pleasing.

3 Advection equation

Solve the advection equation

$$\partial_t f + u\partial_x f = 0 \tag{1}$$

with initial condition $f(x, t = 0) = x$ over all x. Consider fixed boundary condition (i.e., f in the first and the last cells stay fixed in time). Try FTCS and Lax-Friedrich methods and plot them separately on each panel (single window should have two panels; label each panel accordingly). Choose $u = -0.1$ and choose the appropriate timestep and grid spacing so that the Lax-Friedrich scheme is numerically stable.

4 Advection-Diffusion equation

Using the Lax-Friedrich method for advection and the implicit method for diffusion, solve the advection-diffusion equation

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = D\frac{\partial^2 f}{\partial x^2} \tag{2}$$

with the same initial and boundary conditions and u as the previous question. Feel free to copy parts of the code you built in the previous question but this question needs to be submitted as a separate .py file. Choose 2 different diffusion coefficients and plot them on separate panels within the same window. Label each panel according to the assumed diffusion coefficient. Make sure to use the same timestep for both cases (they both need to be numerically stable) so that you can directly compare the evolution of two scenarios.

5 1-Dimensional Hydro solver

Using the donor cell advection scheme, follow the motion of sound waves in a uniform density gas (in 1D, no gravity), starting from some small Gaussian perturbation in density and/or velocity by solving the conservative form of hydro equations:

$$\begin{aligned} \frac{\partial f_1}{\partial t} + \frac{\partial}{\partial x}(u f_1) &= 0 \\ \frac{\partial f_2}{\partial t} + \frac{\partial}{\partial x}(u f_2) &= -\frac{\partial P}{\partial x}. \end{aligned} \tag{3}$$

Plot the density and use reflective boundary conditions. What happens as you increase the amplitude of the perturbation? Do you see a shock? If so, what do you think is setting the width of the shock? (Note: must include written explanations in the README file to receive full credit.)

6 Adiabatic Shock

In the hydro solver above, we have assumed isothermal equation of state (i.e., fixed sound speed), now also track the conservation of energy in the limit of strong perturbation assuming adiabatic process:

$$\frac{\partial f_3}{\partial t} + \frac{\partial}{\partial x}(u f_3) = -\frac{\partial}{\partial x}(uP) \tag{4}$$

where $f_3 = \rho e_{\rm tot}$ is the total energy density. Use the fact that in adiabatic processes

$$\begin{aligned} u &= f_2/f_1 \\ e_{\rm tot} &= f_3/f_1 \\ e_{\rm kin} &= u^2/2 \\ e_{\rm th} &= e_{\rm tot} - e_{\rm kin} \\ P &= (\gamma - 1)\rho e_{\rm th}, \end{aligned} \tag{5}$$

where γ is the adiabatic index. Consider monatomic gas. Verify whether ρ satisfies the shock jump conditions you learned in class.